UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/750,128 | 12/31/2003 | Niniane Wang | 24207-10093 | 9784 |

62296 7590 05/12/2011

GOOGLE / FENWICK
SILICON VALLEY CENTER
801 CALIFORNIA ST.
MOUNTAIN VIEW, CA 94041

| EXAMINER | |
|---|---|
| SCIACCA, SCOTT M | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2478 | |

| NOTIFICATION DATE | DELIVERY MODE |
|---|---|
| 05/12/2011 | ELECTRONIC |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptoc@fenwick.com

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/750,128 | WANG ET AL. | |
| | Examiner | Art Unit | |
| | Scott M. Sciacca | 2478 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *01 March 2011*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-12,15-27,30-41,43 and 44* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12,15-27,30-41,43 and 44* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

This office action is responsive to communications filed on March 1, 2011. Claims 1 and 17 have been amended. New claim 44 has been added. Claims 1-12, 15-27 and 30-41 and 43-44 are pending in the application.

### *Claim Rejections - 35 USC § 103*

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-15, 17-30, 32-41 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hasink et al. (US 2005/0149932) in view of Culbert et al. (US 5,838,968) and Foote (US 7,028,298).

Regarding Claims 1 and 17, Hasink teaches a method comprising:

receiving, by an application executed by an operating system, a plurality of operating parameters having values describing a plurality of resources of a client device (*"Embodiments of the present invention can be used with numerous different operating systems"* – See [0020]; *"an operating system 118, running a foreground process 120 and a background process 122 (such as an index process)"* – See [0051]; *"a background process running at idle priority uses performance counters, optionally including one or more of the counters discussed above, and/or other mechanisms to*

*determine the immediate load on a resource, such as a magnetic or optical mass storage device, it wishes to use"* – See [0024]; *"The determination can take into account the processor or central processing unit (CPU) load, as measured by the time spent in the idle loop, as well as the load on other shared system resources, such as disk drives"* – See [0018]; *"By way of example, the background process checks a performance counter, such as the counter named "\\PhysicalDisk\Current Disk Queue Length" for the specific disk drive instance it wishes to read from or write to. Alternatively or in addition, the background process can access the aggregate total value of the current disk queue lengths for all of the physical disk drives, whose instance is known as "_Total". Advantageously, this is easier than keeping track of which disk drive the process is about to access and checking only that one drive's queue length"* – See [0029]; The queue lengths (operating parameters) of each physical drive (resource) are monitored using the performance counters);

determining a value representing a performance measure of the client device based at least in part on a combination of the plurality of operating parameter values describing the plurality of resources of the client device (*"Alternatively or in addition, the background process can access the aggregate total value of the current disk queue lengths for all of the physical disk drives, whose instance is known as "_Total""* – See [0029]; The queue lengths of each physical drive are combined into the "Total" parameter. Thus, the usage levels of a plurality of resources (physical drives) are aggregated into a single performance counter);

assigning the value representing the performance measure to a usage variable (As mentioned above, the aggregate total of the queue lengths of all the physical disk drives is assigned to the "Total" performance counter); and

correlating by the application a resource usage level of the application with the usage variable and the application modifying its own execution to use a particular resource usage level (*"If the value has changed, the background process uses this as an indication that another process has used the disk in the interim and is possibly still using the disk, and so backs off and waits for an additional period or periods of time"* – See [0037]).

Hasink does not explicitly teach examining a representation of a mapping of usage variable values to resource usage levels, wherein each tuple in the mapping specifies a particular value of the usage variable and a particular resource usage level, identifying a tuple of the mapping for which the particular value of the usage variable matches the value assigned to the usage variable, and the application modifying its own execution to use the particular resource usage level specified by the identified tuple.

In analogous art, Culbert discloses a system for dynamic resource management across tasks that manages a set of resources and optimizes resource allocation (See Abstract). Culbert teaches examining a representation of a mapping of usage variable values to resource usage levels, wherein each tuple in the mapping specifies a particular value of the usage variable and a particular resource usage level (*"In the present embodiment, each task is responsible for deciding the amount of each resource it requires. Referring to FIG. 3, task 350 can specify any number of resource*

*configurations in task resource utilization vector 300. Task resource utilization vectors are located in host memory 123 and are linked together in the host memory. In the present embodiment, task resource utilization vector 300 is coupled to task 350. Tasks must specify at least one task resource utilization record, 310, which specifies the required quantities of the resources managed by resource manager 170 that are necessary for Task 350 to function properly. Task resource utilization record 310, contains an index, 315, indicating the run level associated with this record, and multiple entries 311,312, 314. Each entry initially contains the quantity of the resource requested at this particular level. If a resource entry has no quantity associated with it, i.e. 0, resource manager 170 assumes the task has no requirement for that resource. This does not mean that the task does not use the resource, simply that it does not have a quantifiable need for the resource"* – See Col. 7, lines 41-60);

identifying a tuple of the mapping for which the particular value of the usage variable matches the value assigned to the usage variable (*"FIG. 5 is a block diagram of the degradation method executed by the resource manager of FIG. 1"* – See Col. 4, lines 44-45; *"Referring to FIG. 5, In step 500, the deficit to be filled is determined. A resource deficit can be created by any of the triggering conditions: a new task is going to be created; a task misses a deadline; or an existing task requests more resources"* – See Col. 10, lines 28-32; See also Col. 10, lines 51-67); and

the application modifying its own execution to use the particular resource usage level specified by the identified tuple (*"The task can respond with a quantitative amount*

*of resource it is willing to give up. The task can specify a complete list of resources and amounts it will give up, or just a single resource and amount"* – See Col. 11, lines 1-4).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Hasink to use a mapping of resource usage values to resource usage levels in order to allow an application to modify its execution. Motivation for doing so would be to allow a system programmer to program tasks such that system performance will be dynamically managed and optimized over a programmer-controllable set of system resources (See Culbert, Col. 3, lines 28-45).

Hasink-Culbert does not explicitly teach that the usage variable defines a plurality of usage thresholds for a particular combination of resources of the client device.

However, Foote discloses another resource management system where a usage variable defines a plurality of usage thresholds for a particular combination of resources of the client device (See Col. 4, lines 31-58; A plurality of thresholds are defined (Limit1, Limit2, MaxLimit, etc.) for a particular combination of resources (e.g., CPU, memory, network) in relation to a set of code).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Hasink-Culbert so that a usage variable defines a plurality of usage thresholds for a particular combination of resources of the client device. Motivation for doing so would be to enable resource usage to be tracked and managed for sets of related code. Thus, if a multi-threaded application is using too many resources, the resource usage of the entire application may be throttled back instead of just throttling back individual threads (See Foote, Col. 1, lines 37-51).

Regarding Claims 2 and 18, Hasink teaches the application modifying its own execution comprising the application suspending one or more operations when the value assigned to the usage variable exceeds a threshold (*"When the counter value is non-zero, or greater than a designated threshold, the background process waits a designated amount of time, such as 10 milliseconds, before checking again"* – See [0031]).

Regarding Claims 3 and 19, Hasink teaches the application modifying its own execution comprising the application performing an activity affecting a usage variable proximate to a time that the value assigned to the usage variable indicates an existing activity (*"The background process then waits a given amount of time, such as, by way of example, 10 milliseconds, and checks for pending disk or mass storage I/O by checking the "current disk queue length" counter, or other appropriate performance indicator"* – See [0031]; *"When the counter value is non-zero, or greater than a designated threshold, the background process waits a designated amount of time, such as 10 milliseconds, before checking again"* – See [0031]; The background process (application) will wait a designated amount of time to access a resource (i.e., hard disk) if the resource is already being accessed by another application, before trying to access the resource again).

Regarding Claims 4 and 20, Hasink teaches the application modifying its own execution comprising the application adjusting a rate of operation based at least in part on the value assigned to the usage variable (*"The background process can then determine when idle cycles are being allocated to the background process because another process, such as a foreground process, is waiting for an operation on that same resource to complete. In such cases, the background process optionally refrains from imposing an additional load on the resource, so that the other process can run without delay"* – See [0024]).

Regarding Claims 5 and 21, Hasink teaches the application modifying its own execution comprising the application adjusting a sequence of operations based at least in part on the value assigned to the usage variable (*"An embodiment optionally utilizes a background process which performs indexing of the contents of a user's hard disk without impacting system performance under Windows-NT based operating systems to an extent that would be readily noticeable by a user. The indexing process performs many disk I/O operations when indexing the contents of the user's hard disk to allow the user to rapidly find files which contain certain words, phrases, or strings"* – See [0025]; *"In addition, the index engine can refrain from indexing until it determines that the mass storage device, which stores the data or files to be indexed, is not being utilized by a higher priority or foreground process"* – See [0027]; The sequence of indexing a client device's hard disk is adjusted based on whether or not other higher priority processes

are simultaneously trying to access the hard disk as indicated by the current value of one or more of the performance counters shown in Table 1).

Regarding Claims 6 and 22, Hasink teaches the application modifying its own execution comprising the application adjusting an active feature based at least in part on the value assigned to the usage variable (*"An embodiment optionally utilizes a background process which performs indexing of the contents of a user's hard disk without impacting system performance under Windows-NT based operating systems to an extent that would be readily noticeable by a user. The indexing process performs many disk I/O operations when indexing the contents of the user's hard disk to allow the user to rapidly find files which contain certain words, phrases, or strings"* – See [0025]; *"In addition, the index engine can refrain from indexing until it determines that the mass storage device, which stores the data or files to be indexed, is not being utilized by a higher priority or foreground process"* – See [0027]; The active feature of the background process (application) which is responsible for indexing a client device's hard disk is adjusted when the application refrains from attempting to access the hard drive when other higher priority processes are simultaneously trying to access the hard disk).

Regarding Claims 7 and 23, Hasink teaches the client device (Computer 102 – See Fig. 1) comprising a client processor (CPU 104 – See Fig. 1) and a client memory storage device (Memory 116 – See Fig. 1).

Regarding Claims 8 and 32, Hasink teaches receiving the plurality of operating parameters comprising monitoring at least one of the operating parameters (*"the background process checks a performance counter, such as the counter named "\\PhysicalDisk\Current Disk Queue Length" for the specific disk drive instance it wishes to read from or write to"* – See [0029]).

Regarding Claims 9 and 24, Hasink teaches monitoring a period of inactivity of the client device (*"After the second predetermined time period has elapsed, a determination is made as to whether the computer resource is idle"* – See Abstract).

Regarding Claims 10 and 25, Hasink teaches receiving the plurality of operating parameters comprising receiving at least one of the operating parameters during an initial load of the client processor (*"Embodiments of the present invention determine when a computer and/or resource therein is idle. The determination can take into account the processor or central processing unit (CPU) load"* – See [0018]).

Regarding Claims 11 and 26, Hasink teaches receiving the plurality of operating parameters comprising receiving at least one of the operating parameters during a predetermined time interval (*"The background process checks the value of this counter before and after an interval, such as the 10 millisecond wait interval described above"* – See [0037]).

Regarding Claims 12 and 27, Hasink teaches the plurality of operating parameters comprising a client processor load (*"Embodiments of the present invention determine when a computer and/or resource therein is idle. The determination can take into account the processor or central processing unit (CPU) load"* – See [0018]).

Regarding Claims 15 and 30, Hasink teaches the method of Claim 7 further comprising writing to a computer readable medium of the client memory storage device (*"while running, the indexing process is constantly reading from and writing to the user's hard disk"* – See [0009]).

Regarding Claim 33, Hasink teaches the usage variable being a quantitative performance measure of the client device (Table 1 shows the various counters that may be monitored. Note that the counters shown in Table 1 are quantitative performance measurements, such as "% idle time" or "Disk Bytes/sec").

Regarding Claim 34, Hasink teaches the usage variable being a qualitative performance measure of the client device (*"the background process checks a performance counter, such as the counter named "\\PhysicalDisk\Current Disk Queue Length" for the specific disk drive instance it wishes to read from or write to"* – See [0029]; *"a check of the "current disk queue length" performance counter may not be, on its own, adequate or sufficient to allow a background process to determine whether or*

*not another process is using the disk drive, because a queued operation might be on behalf the background process itself"* – See [0030]; One performance counter shown in Table 1 is "Current Disk Queue Length". While this value is a number, it does not directly and numerically indicate a performance measure).

Regarding Claim 35, Hasink teaches the application modifying its own execution comprising the application throttling back its usage of the client device (*"The background process can then determine when idle cycles are being allocated to the background process because another process, such as a foreground process, is waiting for an operation on that same resource to complete. In such cases, the background process optionally refrains from imposing an additional load on the resource, so that the other process can run without delay"* – See [0024]).

Regarding Claim 36, Hasink teaches the application dynamically modifying its own execution based on dynamic changes to the value assigned to the usage variable (*"If the value has changed, the background process uses this as an indication that another process has used the disk in the interim and is possibly still using the disk, and so backs off and waits for an additional period or periods of time, such as additional 10 millisecond intervals, until the counter value stops changing"* – See [0037]).

Regarding Claim 37, Hasink teaches the application modifying its own execution comprising the application pausing between execution of resource-intensive

calculations (*"at state 316 the background process waits a designated period of time, such as 10 msec. At state 318, a determination is then made as to whether the disk is in use"* – See [0054]).

Regarding Claim 38, Hasink teaches a resource used by the application being memory (Memory 116 – See Fig. 1) and wherein the application modifying its own execution comprises the application dynamically scaling back its memory usage based on dynamic changes to the value assigned to the usage variable (The example given above deals with the background process (application) modifying its own execution with regard to accessing one or more hard disks. Hard disks are a type of memory and the usage of the hard disk by the application includes performing "seeks" for data on the hard disk during the indexing procedure (also mentioned above)).

Regarding Claim 39, Hasink teaches a resource used by the application being network bandwidth (*"Similarly, the above techniques can be applied to a shared network with limited bandwidth"* – See [0050]) and wherein the application modifying its own execution comprises the application throttling-back usage of network bandwidth based on dynamic changes to the value assigned to the usage variable (*"there may be multiple processes trying to access the Internet, and use of the foregoing techniques avoid having a background process slow down a transfer being made by a foreground process"* – See [0050]).

Regarding Claim 41, Hasink teaches a plurality of usage variables (See Table 1) and wherein the correlating comprises the application modifying its own execution based at least in part on changes to values assigned to the plurality of usage variables (*"In an example embodiment, a background process running at idle priority uses performance counters, optionally including one or more of the counters discussed above, and/or other mechanisms to determine the immediate load on a resource, such as a magnetic or optical mass storage device, it wishes to use"* – See [0024]).

Regarding Claims 40 and 43, Culbert teaches the representation of the mapping comprising a table having a column representing the usage variable and a column representing the resource usage level of the application and each tuple in the mapping corresponds to a row in the table (*"In the present embodiment, each task is responsible for deciding the amount of each resource it requires. Referring to FIG. 3, task 350 can specify any number of resource configurations in task resource utilization vector 300. Task resource utilization vectors are located in host memory 123 and are linked together in the host memory. In the present embodiment, task resource utilization vector 300 is coupled to task 350. Tasks must specify at least one task resource utilization record, 310, which specifies the required quantities of the resources managed by resource manager 170 that are necessary for Task 350 to function properly. Task resource utilization record 310, contains an index, 315, indicating the run level associated with this record, and multiple entries 311,312, 314. Each entry initially contains the quantity of the resource requested at this particular level. If a resource*

*entry has no quantity associated with it, i.e. 0, resource manager 170 assumes the task has no requirement for that resource. This does not mean that the task does not use the resource, simply that it does not have a quantifiable need for the resource"* – See Col. 7, lines 41-60).

Claim 44 is rejected based on reasoning similar to Claim 1.

3. Claims 16 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hasink et al. (US 2005/0149932) in view of Culbert et al. (US 5,838,968) and Foote (US 7,028,298) and further in view of Anderson, II et al. (US 5,909,544).

Regarding Claims 16 and 31, Hasink does not explicitly teach the plurality of operating parameters comprising a first parameter and a second parameter, wherein the first parameter comprises a speed of the client processor and the second parameter comprises a capacity of the client memory storage device.

However, Anderson does teach the operating parameter comprising a first parameter and a second parameter, the first parameter comprising a speed of the client processor and the second parameter comprising a capacity of the client memory storage device (*"It is an object of the invention to provide a system for tracking and scheduling of available resource computers connected in a network, including monitoring such parameters as, for example, the location, name, operating system, memory, speed, processor characteristics, memory capacity and other operational*

*characteristics, of each resource computer, and using that information to allocate those resource computers to run applications, such as for example, test applications and collect data, such as test data"* – See Col. 4, lines 22-30).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include processor speed and storage capacity as operating parameters. One of ordinary skill would have been motivated to do so since Anderson shows in Col. 4, lines 22-30 that processor speed and memory capacity are among several parameters that are important to take into consideration when allocating resources.

### *Response to Arguments*

4. Applicant's arguments with respect to Claims 1, 17 and 44 have been considered but are moot in view of the new grounds of rejection.

### *Conclusion*

5. Applicant's amendment necessitated the new grounds of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Scott M. Sciacca whose telephone number is (571) 270-1919. The examiner can normally be reached on Monday thru Friday, 7:30 A.M. - 5:00 P.M. EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeff Pwu can be reached on (571) 272-6798. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Scott M. Sciacca/
Examiner, Art Unit 2478

/Jeffrey Pwu/
Supervisory Patent Examiner, Art Unit 2478